参考摘抄于：<https://es6.ruanyifeng.com/#docs/module-loader>

**1、浏览器加载ES6模块**

使用script标签，需加入type=”module”属性。



浏览器对于带有type="module"的<script>，都是异步加载，不会造成堵塞浏览器，即等到整个页面渲染完，再执行模块脚本，等同于打开了<script>标签的defer属性。



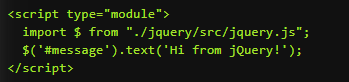
如果网页有多个<script type="module">，它们会按照在页面出现的顺序依次执行。

<script>标签的async属性也可以打开，这时只要加载完成，渲染引擎就会中断渲染立即执行。执行完成后，再恢复渲染。

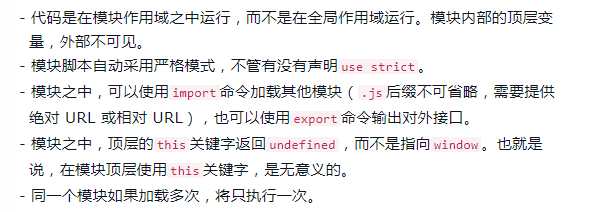


一旦使用了async属性，<script type="module">就不会按照在页面出现的顺序执行，而是只要该模块加载完成，就执行该模块。

ES6 模块也允许内嵌在网页中，语法行为与加载外部脚本完全一致。



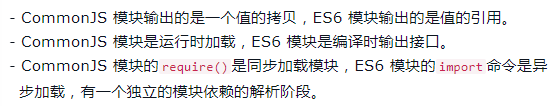
对于外部的模块脚本（上例是foo.js），有几点需要注意。



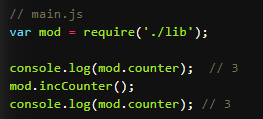
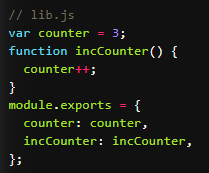
利用顶层的this等于undefined这个语法点，可以侦测当前代码是否在 ES6 模块之中。

**2、ES6 模块与 CommonJS 模块的差异**

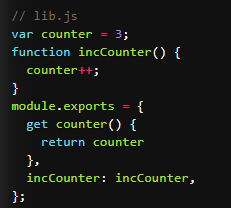
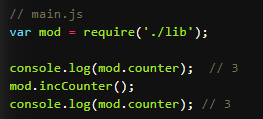
ES6 模块与 CommonJS 模块的差异有以下几点：



CommonJS 模块输出的是值的拷贝，也就是说，一旦输出一个值，模块内部的变化就影响不到这个值。



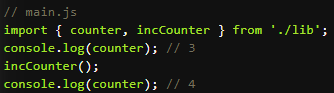
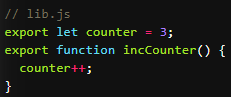
上面代码说明，lib.js模块加载以后，它的内部变化就影响不到输出的mod.counter了。这是因为mod.counter是一个原始类型的值，会被缓存。除非写成一个函数，才能得到内部变动后的值。



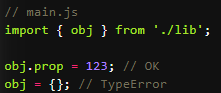


上面代码中，输出的counter属性实际上是一个取值器函数。现在再执行main.js，就可以正确读取内部变量counter的变动了。

ES6 模块的运行机制与 CommonJS 不一样。JS 引擎对脚本静态分析的时候，遇到模块加载命令import，就会生成一个只读引用。等到脚本真正执行时，再根据这个只读引用，到被加载的那个模块里面去取值。换句话说，ES6 的import有点像 Unix 系统的“符号连接”，原始值变了，import加载的值也会跟着变。因此，ES6 模块是动态引用，并且不会缓存值，模块里面的变量绑定其所在的模块。

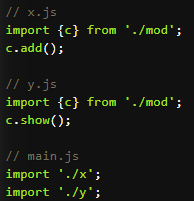
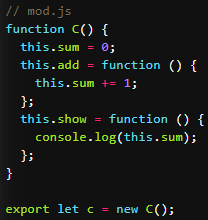


上面代码说明，ES6 模块输入的变量counter是活的，完全反应其所在模块lib.js内部的变化。由于 ES6 输入的模块变量，只是一个“符号连接”，所以这个变量是只读的，对它进行重新赋值会报错。



上面代码中，main.js从lib.js输入变量obj，可以对obj添加属性，但是重新赋值就会报错。因为变量obj指向的地址是只读的，不能重新赋值，这就好比main.js创造了一个名为obj的const变量。

export通过接口，输出的是同一个值。不同的脚本加载这个接口，得到的都是同样的实例。



上面代码说明了x.js和y.js加载的都是C的同一个实例。

**3、Node.js 的模块加载方法**

JavaScript 现在有两种模块。一种是 ES6 模块，简称 ESM；另一种是 CommonJS 模块，简称 CJS。CommonJS 模块是 Node.js 专用的，与 ES6 模块不兼容。语法上面，两者最明显的差异是，CommonJS 模块使用require()和module.exports，ES6 模块使用import和export。

它们采用不同的加载方案。从 Node.js v13.2 版本开始，Node.js 已经默认打开了 ES6 模块支持。Node.js 要求 ES6 模块采用.mjs后缀文件名。也就是说，只要脚本文件里面使用import或者export命令，那么就必须采用.mjs后缀名。Node.js 遇到.mjs文件，就认为它是 ES6 模块，默认启用严格模式，不必在每个模块文件顶部指定"use strict"。如果不希望将后缀名改成.mjs，可以在项目的package.json文件中，指定type字段为module。



总结：.mjs文件总是以 ES6 模块加载，.cjs文件总是以 CommonJS 模块加载，.js文件的加载取决于package.json里面type字段的设置。

Node.js 规定 ES6 模块之中不能使用 CommonJS 模块的特有的一些内部变量。首先，就是this关键字。其次，以下这些顶层变量在 ES6 模块之中都是不存在的。